ABSTRACT OF THE DISCLOSURE

Methods of instrumenting synthesizable source code to enable debugging support akin to high-level language programming environments for gate-level simulation are provided. One method of facilitating gate-level simulation includes generating cross-reference instrumentation data including instrumentation logic indicative of an execution status of at least one synthesizable register transfer level (RTL) source code statement. A gate-level netlist is synthesized from the source code. Evaluation of the instrumentation logic during simulation of the gate-level netlist facilitates simulation by indicating the execution status of a corresponding source code statement. One method results in a modified gate-level netlist to generate instrumentation signals corresponding to synthesizable statements within the source code. This may be accomplished by modifying the source code or by generating the modified gate-level netlist as if the source code was modified during synthesis. Alternatively, cross-reference instrumentation data including instrumentation logic can be generated without modifying the gate-level design. The instrumentation logic indicates the execution status of a corresponding cross-referenced synthesizable statement. An execution count of a cross-referenced synthesizable statement can be incremented when the corresponding instrumentation signals indicates the statement is active to determine source code coverage. Source code statements can be highlighted when active for visually tracing execution paths. For breakpoint simulation, a breakpoint can be set at a selected source code statement. The corresponding instrumentation logic from the cross-reference instrumentation data is implemented as a simulation breakpoint. The simulation is halted at a

simulation cycle where the values of the instrumentation signals indicate that the source code statement is active.